AMENDMENTS TO THE SPECIFICATION:

Please replace paragraph **[0004]** with the following rewritten paragraph:

[0004] A motion ~~estimation~~ detector 10 needs a lot of memory accesses and a large memory size in implementing a MPEG-4 algorithm and greatly influences ~~on~~ the performance of the MPEG-4 video encoder algorithm. There is a kind of pictures that is P-frame or B-frame, according to the frame direction of motion estimation. However, their motion estimations are similar to each other in that they estimate motion vector by using memory access of a current macro-block and a reference search area of a previous or next frame. So, only the motion estimation unit for P-frames will be described in this specification. The motion ~~estimation~~ detector 10 for P-frames estimates motion vectors by using previous frame (t-1) and the current frame (t) as shown in FIG. 2A. In other words, the motion ~~estimation~~ detector 10 finds the most similar location by moving a macro-block of the current frame on a reference search area of a previous frame by pixel. A mean square error (MSE) method, a sum of absolute difference (SAD) method and a mean absolute difference (MAD) method are proposed as the methods to find similarity of a current macro-block and a reference search area.

Please replace paragraph **[0005]** with the following rewritten paragraph:

[0005] FIGs. 2A and 2B ~~illustrate an~~ illustrate a conventional implementation of a motion estimation method according to an MPEG-4 video encoding algorithm. In the conventional motion estimation method shown in FIG. 2A, a reference frame and a current frame are stored in an external memory to find similarity. The higher frame resolution the video has, the more memory the method requires. Accordingly, when an internal memory has a small capacity, an external memory should be used since the internal memory cannot store all the data. When an external memory is used, the system implementing the method becomes slower, since the external memory is slower to be accessed compared with internal

memory within a microprocessor. Furthermore, since frequent access to the external memory consumes a large amount of electric power, the method could not be applied to a mobile terminal which attaches importance to power consumption to ~~use battery long~~ use a battery for a long time.

Please replace paragraph **[0023]** with the following rewritten paragraph:

[0023] FIG. 9 illustrates a relationship ~~relation~~ between a ~~macro lock~~ macro block and buffer allocation for a reference search region when the motion detection method according to an embodiment of the present invention is executed at the reference search range [-8, 7]; and

Please replace paragraph **[0027]** with the following rewritten paragraph:

[0027] As shown in FIG. 3, in the motion estimation method of the mobile terminal according to the present invention, it is possible to increase the number of buffers instead of reducing the size of the buffer, and reduce the size of the internal memory used in the conventional double buffering method, so as to more effectively use a small internal memory. In other words, in the ~~convention~~ conventional double buffering method shown in FIG. 4A, nine macro blocks are used and discarded when detecting motion of one macro block for the reference search area of the internal memory. However, as shown FIG. 4B, in the method suggested by the present invention, one buffer P[3] consisting of three macro blocks is further provided to use a circular buffering configuration. Only three macro blocks are moved in memories to transfer data between memories so that it is possible to detect motion.

Please replace paragraph **[0029]** with the following rewritten paragraph:

[0029] ~~Considering motion estimation method~~ Considering the motion estimation method shown in FIG. 3 in the aspect of the frame, there are two methods. In the

first method, the motion of the macro block of the frame proceeds first vertically, and then is initialized and proceeds horizontally as shown in FIG. 5A. In the second method, the motion of the macro block of the frame proceeds first horizontally and is initialized and proceeds vertically as shown in FIG. 5B. FIG. 5C illustrates buffer allocation and data transfer of the cases that motion is detected vertically. FIG. 5D illustrates buffer allocation and data transfer of the cases that motion is detected horizontally. The value "t" indicates the location of the macro block desired to detect motion.

Please replace paragraph **[0031]** with the following rewritten paragraph:

[0031] First, as shown in FIG. 5A, the vertical motion estimation is performed according to the flowchart of FIG. 6. In other words, if a motion detection algorithm begins for a frame (S111), the numbers of macro blocks of a frame are obtained horizontally and vertically from the numbers of pixels of width and height of the frame, and the horizontal and vertical initial values of macro blocks are set to be "0" (S111). At a first start point or vertical start point, data in the three buffers P[0], P[1] and P[2] bounded in three macro blocks are brought from the external memory and stored in the internal memory (S112). The three above-mentioned buffers are the reference search area of the first macro block of the current frame, and the motion of the macro block is detected within the reference search area (S114). When the motion estimation is completed on one macro block, the vertical number of the macro blocks is counted up by one to detect motion of the next macro block vertically (S115). Then, it is ascertained whether the macro block is the last macro block in a vertical direction of the current frame (S116). At the step (S116), if the macro block is not the last macro block in a vertical direction of the current frame, three macro block data for the reference search area are transferred to the buffers located at P[{(the vertical number of macro blocks)+2} % 4] to detect the motion of the macro block in the next column, wherein the symbol "%" refers to a modulo division operation, as known

to those knowledgeable in the field of software engineering. If it is determined that the motion detection is completed on blocks in a column at the step (S116), the value of column is initialized and the value of row is counted up by one (S117). Now, it is determined whether the motion detection is completed in a horizontal direction by determining whether the block in a row is terminated (S118). If the motion detection is not completed in the horizontal direction at the step S118, it goes to the step (S118) to continuously perform the motion estimation algorithm. If the motion detection is completed in the horizontal direction at the step (S118), the motion estimation algorithm is terminated (S119).

Please replace paragraph **[0035]** with the following rewritten paragraph:

[0035] Equation 1

For ~~$0 \leq j < 16$,~~ $-16 \leq j < 0$, if the desired macro block exists between $P_{ref}[(x-1)\%4]$ and $P_{ref}[x\%4]$,

$$\begin{cases} SAD_1 = \sum_{k=0}^{15-j}\sum_{l=0}^{15}\left|P_c[k][l] - P_{ref}[(x-1)\%4][k+j][l]\right| \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ 13a \\ SAD_2 = \sum_{K=15-J}^{15}\sum_{L=0}^{15}\left|P_c[k][l] - P_{ref}[x\%4][k-(15-j][l]\right| \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ 13b \end{cases}$$

For ~~$16 \leq j < 32$,~~ $0 \leq j < 15$, if the desired macro block exists between $P_{ref}[x\%4]$ and $P_{ref}[(x+1)\%4]$,

$$\begin{cases} SAD_1 = \sum_{k=0}^{15-j}\sum_{l=0}^{15}\left|P_c[k][l] - P_{ref}[x\%4][k+j][l]\right| \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ 13a \\ SAD_2 = \sum_{k=15-j}^{15}\sum_{l=0}^{15}\left|P_c[k][l] - P_{ref}[(x+1)\%4][k-(15-j)][l]\right| \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ \Lambda\ 13b \end{cases}$$

Please replace paragraph **[0038]** with the following rewritten paragraph:

[0038] Equation 2

For ~~$0 \le j < 16$,~~ <u>$-16 \le j < 0$</u>, if the desired macro block exists between $P_{ref}[(x+1)\%4]$ and $P_{ref}[x\%4]$,

$$\begin{cases} SAD_1 = \sum_{k=0}^{15}\sum_{l=0}^{15-i}\left|P_c[k][l] - P_{ref}[(x-1)\%4][k]][l+i]\right| \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ 13a \\ SAD_2 = \sum_{k=0}^{15}\sum_{l=15-i}^{15}\left|P_c[k][l] - P_{ref}[x\%4][k][l-(15-i)]\right| \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ 13b \end{cases}$$

For $16 \le i < 32$, if the desired macro block exists between $P_{ref}[x\%4]$ and $P_{ref}[(x+1)\%4]$,

$$\begin{cases} SAD_1 = \sum_{k=0}^{15}\sum_{l=0}^{15-i}\left|P_c[k][l] - P_{ref}[x\%4][k][l+i]\right| \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ 13a \\ SAD_2 = \sum_{k=0}^{15}\sum_{l=15-i}^{15}\left|P_c[k][l] - P_{ref}[(x+1)\%4][k][l-(15-i)]\right| \Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ 13b \end{cases}$$